Welcome!

Course Instructor

- Name: Dr. Jianwei Niu
- Office: SB 4.01.36
- Office Phone: 458-7360
- Email: niu@cs.utsa.edu

Course Meetings, Web Pages, etc.

- Course Meetings: MW 4:00 - 5:15 in classroom UTSA-1604 HSS 3.04.06
- Office Hours: Wednesday 10:00 - 11:30
- Course Web Page: http://www.cs.utsa.edu/~niu/teaching/cs3703Spr07
Grading Scheme

- Midterm: 15%
- Final: 35%
- Assignments: 10%
- Projects: 35%
- Participation: 5%

Course Textbooks


Course Topics

- Software Development Process and Project Management
- Requirements Analysis
- Unified Modeling Language (UML) and Other Specification Notations
- Software Architecture Design
- Testing Techniques

What Is Software

- Software is a collection of artifacts
  - Computer programs
  - Data
  - Documents
- Characteristics of software
  - Software is either customized product or generic product
  - Software is a logical system
  - Software evolves
What Is Software

 Attribute of good software
 - Dependability
   availability, reliability, security, and safety
 - Efficiency
   processing time, memory utilization, responsiveness,
 - Usability
   appropriate user interface and adequate documentation
 - Maintainability
   ease of change

What Is Software Engineering

 [Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines

 by Prof. Fritz Bauer at the 1968 NATO conference on software technology, in Garmisch, Germany.

 In short, software engineering is about developing quality software in a predictable way.

 Key phrases: engineering discipline, all aspects of software production

What Is Software Engineering

 Software engineering is a discipline that integrates
 - Process
   provides a framework for software development
 - Methods
   provide "how to's" for building software
 - Tools
   provide automated or semi-automated support for the process and the methods

Software Process

 Software process is a set of activities used for the development of software systems
 - Communication
 - Planning
 - Modeling
 - Construction
 - Deployment
Each activity is accomplished by a collection of tasks, such as specification, analysis, and validation.

A number of umbrella activities:
- Software project tracking and control
- Risk management
- Software quality assurance
- Formal technical reviews
- Measurement
- Software configuration management
- Reusability management
- Work product preparation and production

A process model describes:
- What steps you go through
- Which development artifacts are produced, and when
- How activities are coordinated

Different process models:
- The waterfall model
- The prototyping model
- The iterative model
- The incremental model
- Others: RUP, agile methods, component-based SE, and ...

The classic lifecycle
- Sequential: each stage completed before the next started
- Mainly used by large companies
- Problems
  - But real projects are rarely sequential
  - Customer has to wait for product until the end
  - Hard to state requirements up-front
  - Heavy documentation
**Requirements engineering**

The Incremental Model

- Development done in defined cycles
- All requirements analysis is done in one phase
- Each increment produces deliverable
- Still has the problem of having to elicit all the requirements up front

The Prototype Model

- Using prototype for discovering requirements
- Particularly good for user interfaces
- Problems
  - Customer doesn't like to have to "rebuild" (have to agree to discard it)
  - Make implementation decisions that you stick with
The Iterative Model

- Requirements engineering
- Design and implementation
- Testing

Multiple mini-waterfall-like cycles
- A working implementation in each iteration as in incremental; no throwaways as in prototype
- Requirements gathering spreading out over the iterations
- Usually with less non-essential artifacts produced
- Rational Unified Process (RUP): an example of the iterative model specifically for UML

The Rational Unified Process Model

- Inception: business case establishment
  - Vision
  - Scope
  - Project feasibility
- Elaboration: problem understanding
  - Requirements model
  - Architecture design
  - Risk identification and resolution
- Construction: implementation
- Transition: deployment
The Rational Unified Process Model

- It is a hybrid process model
  - Elements from different models
  - Iteration in each phase and in the whole process
- It supports multiple views
  - Dynamic view
  - Static view
  - Practice view
- Requirements gathering spreading out over the iterations

Software Process Models

- Every software development group has its own process model(s)
- They vary along several dimensions:
  - Formal vs. ad hoc
  - Sequential vs. concurrent
  - Lots vs. little documentation
  - …